In-class Participation Assignment

Group 16

MIE 1624H Due: November 15, 2022

Introduce `scikit-learn`, `nltk` and other libraries in Python in the context of working with text data. Explain basic feature extraction from text data (number of words, number of characters, average word length, number of stopwords, etc.) and basic text preprocessing (lower casing, punctuation and special characters removal, stopwords removal, rare words removal, spelling correction, tokenization, stemming, lemmatization, etc.). Explain advanced text processing (N-grams, Term Frequency (TF), Inverse Document Frequency (IDF), Term Frequency-Inverse Document Frequency (TF-IDF), "Bag of Words" document representation, word embedding, etc.).

Twitter provides an application programming interface (API) for Python to extract data about tweets. Briefly present what steps do you need to take to get Twitter data in Python. Demo what Python modules and data structures can be used to efficiently work with Twitter data. Explain limitations of the Twitter API. Discuss alternative ways, if exist, to get Twitter data in Python. In your IPython example, use Twitter API for Python to download tweets, search tweets by hashtags, extract metadata such as a number of retweets, etc. Use Twitter API for Python to download tweets and save those as a csv file. In your IPython example, also perform basic feature extraction and basic text preprocessing on tweets from your csv file.

Many web-portals provide an application programming interface (API) in Python to extract data from news web-sites. Other news web-sites provide RSS feeds for extracting news data for future analysis. Use one of RSS feed parsing libraries in Python such as `Feedparser` to work with one of the news web-sites. Extract most commonly used elements in RSS feeds such as "title", "link", "description", "publication date", "entry ID", etc. Follow "links" in the RSS feed and extract text of news articles using web scraping in Python or other Python methods. You can use `BeautifulSoup` library in Python for pulling data out of HTML and XML files. In your IPython example, use a news web-site of your choice and create a Pandas dataframe from your RSS parsing results representing fields of metadata from that article, e.g., title, date, link, author, etc., as well as text of the news article. Print a table with your results and export those as a `csv` file with each row as an article and columns as article's metadata (including full text of each article). Modify your code and show your results for 4-5 commonly used news web-sites of your choice such as New York Times, Globe and Mail, etc. In your IPython example, also perform advanced text processing that you explained, using news articles that you extracted from one of news web-sites. Explain "Bag of Words" document representation in Python based on both TF and TF-IDF, and prepare appropriate data structures in Python to be used by machine learning algorithms of `scikit-learn` package. Explain word embedding techniques and prepare appropriate data structures in Python to be used by machine learning algorithms of `scikit-learn` package.

Prepare 10 minute presentation of your results. Before the presentation, upload your PowerPoint slides, PDF slides, IPython Notebook `ipynb` file(s) and all data files in `zip` archive via Quercus portal, such that those can be posted on the course web-page and re-used by your colleagues for assignments and a course project. Presentation materials should be uploaded to Quercus portal by 4:00pm on the due date. If you have any questions about your in-class presentation assignment, please contact course TAs Eric Floro `eric.floro@mail.utoronto.ca` or Saeede Hasanpoor `saeede.hasanpoor@utoronto.ca`.